METR4202 Robotics & Automation
Semester 2, 2023
Tutorial 5

Motion and Path Planning II: Sampling-based Planning

Your internship at Australia's largest automated warehousing and logistics company Murray-Darling is going well. After your success last week in developing the company's country-wide path planning framework, your supervisor has now assigned you the new task of designing the motion planning algorithms for the automated ground robots that operate in the company's factories. A typical factory floorplan is shown in Figure 1.

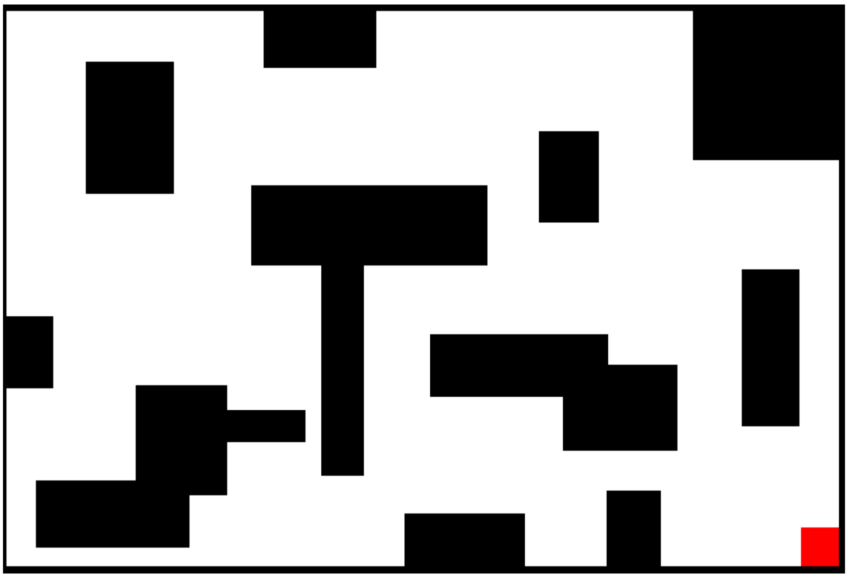

Figure 1: Typical autonomous warehouse floorplan at the Murray-Darling. The company's automated ground robots need to navigate from various points across this space to reach the pick-up and drop-off location shown in red (bottom right).

1 Probabilistic Roadmaps

Exercise 1.1. Assuming that the warehouse robots are holonomic in the 2D plane, what is the configuration space of the planning problem?

Exercise 1.2. Implement the PRM* algorithm to automatically generate a roadmap given a floorplan. As a starting point, you've been provided with skeleton code in `prm_graph_generation.m`. This program will read floorplans stored as .bmp files in the `/maps` folder and will output the generated roadmap as `*_graph.txt` and `*_coords.txt` files, which are compatible with your graph search programs from Tutorial 4.

Remember that the PRM* connection radius is computed as:

$$r = \gamma_{PRM} \left(\frac{\log n}{n} \right)^{\frac{1}{d}},$$

where γ_{PRM} must satisfy the condition:

$$\gamma_{PRM} > 2\left(1+\frac{1}{d}\right)^{\frac{1}{d}}\left(\frac{\mu\left(X_{free}\right)}{\zeta_d}\right)^{\frac{1}{d}}.$$

Note that n is the number of samples, d is the dimensionality of the problem, $\mu\left(X_{free}\right)$ is the free space area, and ζ_d is the volume of the unit ball in d-dimensional Euclidean space.

You've also been provided with the helper function `in_collision.m`, which accepts a vector of points and an occupancy map and returns true if any of the points are in collision; it returns false otherwise.

Exercise 1.3. Change the number of sampled points n to see how the generated graph changes. In particular, note the behaviour in narrow passages, the number of connected components as well as the computation time.

2 Rapidly Exploring Random Trees

Exercise 2.1. Your supervisor has informed you that they are introducing differential drive robots into the warehouses. Why are the PRMs generated in the previous question no longer suitable for planning with the differential drive robots?

Exercise 2.2. You decide that perhaps an RRT planner would be a better option as you can easily adapt it to the steering strategy used by each robot. Starting from the skeleton code provided in `rrt.m`, implement the RRT algorithm. You are provided with several helper functions:

- `diff_drive_steer.m`: simulated the motion of a 2D differential drive point robot
- `holonomic_steer.m`: simulates the motion of a 2D holonomic point robot
- `in_region.m`: checks if a 2D point lies within a 2D rectangular region
- `plot_rrt.m`: plots the RRT

See the individual functions for full I/O documentation.

You can select from the two `*_steer.m` functions by assigning the function handle on line 32 of `rrt.m`.

An example RRT generated for a differential drive robot is shown in Figure 2

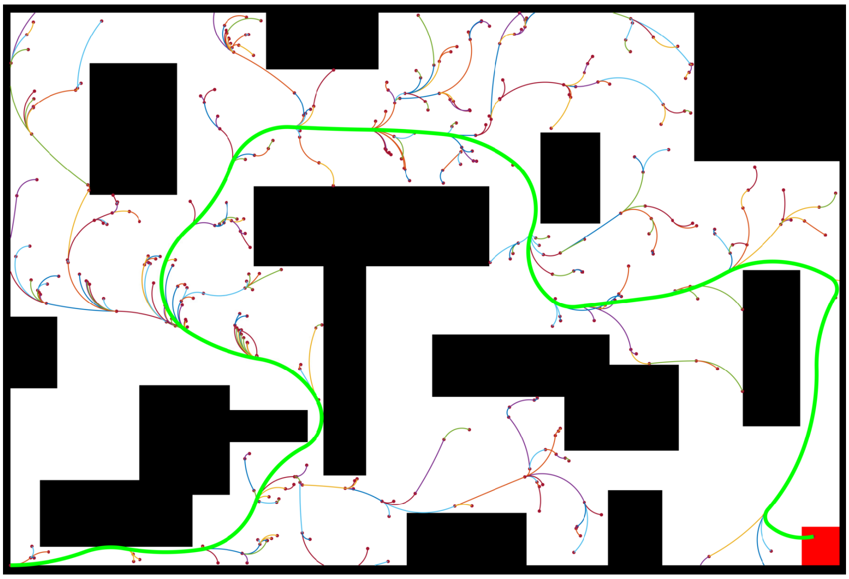

Figure 2: Example output from the RRT planner.

Exercise 2.3. Run your RRT planner with the different steering functions and different random seeds to see how the path changes. In particular, note evidence of suboptimality in the tree expansion. Describe how RRT* enables asymptotic optimality.

Exercise 2.4. Describe one other sampling strategy that has been used with RRTs to improve path search efficiency.